



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Teemu Antti Matias Kuusela

RentéFin

videovuokrausohjelmisto

Tekniikka ja liikenne

2010

ALKUSANAT

Tämä opinnäytetyö on tehty Vaasan ammattikorkeakoulun tietotekniikan koulutusohjelman päättötyönä kevään ja syksyn 2009 aikana. Työ tehtiin projektityönä kauhajokiselle SIP-Viihde Oy:lle, joka paremmin tunnetaan nimellä Viihdestoppi. Työn kirjallinen osuus kirjoitettiin kevään 2010 aikana.

Viihdestopin yhteyshenkilönä toimi myymäläpäällikkö Ilmo Kujansivu. Vaasan ammattikorkeakoulun puolesta työn valvojana toimi yliopettaja Pirjo Prosi.

Kiitokset haluan esittää Viihdestopin omistajille ja koko henkilökunnalle sekä muille työn teossa minua auttaneille.

Vaasassa 30.4.2010

Teemu Kuusela

VAASAN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Teemu Kuusela
Opinnäytetyön nimi	RentéFin videovuokrausohjelmisto
Vuosi	2010
Kieli	suomi
Sivumäärä	43 + 2 liitettä
Ohjaaja	Pirjo Prosi

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa Viihdestopille uusi videovuokrausohjelmisto alkuperäisen ohjelman tilalle. Opinnäytetyönä toteutettu ohjelmisto pitää sisällään kokonaisvaltaiset elokuvien vuokraus- ja palautustoiminnot sekä runsaasti muita toimintoja tukevia ominaisuuksia.

Ohjelmisto toteutettiin käyttäen Java-ohjelmointikieltä ja tietojen varastointiin käytetään MySQL-tietokantojen hallintajärjestelmää. Lisäksi ohjelmisto pitää sisällään verkkosivuilla sijaitsevan hakuominaisuuden, jonka avulla asiakas pystyy hakemaan tietokannasta tietoja yrityksen elokuvista ja niiden läsnäoloista. Aineistona käytettiin alaan liittyvää kirjallisuutta sekä verkosta löytyviä käytössä olevien ohjelmien spesifikaatioita.

Työn tuloksena syntynyt ohjelmisto pitää sisällään kaikki vaatimusmäärittelyssä esiintyvät ominaisuudet. Työn kirjallista osuutta kirjoitettaessa valmis ohjelma ei kuitenkaan ole vielä yrityksen käytössä, mutta se tullaan ottamaan käyttöön kun palvelimeen liittyvät seikat on varmistettu.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietotekniikan koulutusohjelma

ABSTRACT

Author	Teemu Kuusela
Title	RentéFin video renting software
Year	2010
Language	Finnish
Pages	43 + 2 Appendices
Name of Supervisor	Pirjo Prosi

The objective of the thesis was to engineer and produce a new video rental software for Viihdestoppi as a replacement for their original one. The engineered software includes movies overall renting and returning functions with multiple other features to support them.

The software accomplished using Java-programming language and MySQL database management system for data storage. The software contains a search feature in the Viihdestoppi's web page, which the client can use to search from database for information of the movies and about their presence. For study material I have used literature about the relevant subject and specifications about techniques used.

The finished application contains every feature from the requirement analysis. During the writing process the software is not yet in use but it will be deployed as features relates to server has been confirmed.

Keywords Programming, Database, MySQL, Java

LYHENNELUETTELO

CMD	Command Prompt Komentorivi tulkki
HTML	Hypertext Markup Language Verkkosivuissa käytettävä kuvauskieli
HTTPd	Hypertext Transfer Protocol – demon Apachen selaimen tiedonsiirtoprotokollaa käyttävä palvelin
JDBC	Java Database Connectivity Ohjelmointirajapinta tietokannan ja Java-sovelluksen välillä
JDK	Java Standard Edition Development Kit Java-ohjelmien kehitysympäristö
JRE	Java Standard Edition Runtime Environment Java ohjelmien suoritusympäristö
JVM	Java Virtual Machine, Tulkki, eli virtuaalikone joka suorittaa Java-ohjelmat

MySQL	My Structured Query Language Palvelinperustainen relaatiotietokantojen hallintajärjestelmä
PHP	Hypertext Preprocessor Verkkosovellusten ohjelmointikieli
SDK	Software Development Kit Kokoelma kehitystyökaluja luokkien luontiin
SQL	Structured Query Language Standardoitu tietokantojen kyselykieli
TDD	Test-Driven Development Testivetoinen ohjelmistokehitys
UTF	Unicode Transformation Format Yleismaailmallinen tietokonejärjestelmien merkistöstandardi
WWW	World Wide Web Maailmanlaajuinen Internet-verkko

SISÄLLYS

ALKUSANAT	2
TIIVISTELMÄ	3
ABSTRACT	4
LYHENNELUETTELO	5
1 JOHDANTO	9
1.1 Tavoitteet	9
1.2 Tietoja asiakasyrityksestä	9
2 PROJEKTIN KUVAUS	11
2.1 Yleiskuvaus	11
2.2 Käytettävyyteen liittyvät määritelmät	11
2.3 Lisäominaisuudet	12
3 KÄYTETYT OHJELMISTOT JA TEKNIIKAT	13
3.1 Ohjelmointiympäristö	13
3.2 Ohjelmointitekniikat	13
3.3 Tietokannat	15
3.4 Palvelin	16
4 TOIMINTOJEN KUVAUS	18
4.1 Yleiskuvaus	18
4.2 Toiminnot	19
4.2.1 Vuokraus ja palautus	19
4.2.2 Tietojen lisäys ja muokkaus	23
4.2.3 Lisäominaisuuksia	24
4.3 Tiedot ja tietokannat	25
5 TEKNINEN MÄÄRITTELY	31
6 TOTEUTUS	32
6.1 Pääohjelma	32
6.2 Lisäosa verkkosivuille	34
6.3 Kohdatut ongelmat	37

7	TESTAUS.....	39
	7.1 Johdanto.....	39
	7.2 Ympäristö.....	39
8	YHTEENVETO	41
	LÄHTEET.....	42
	LIITTEET	43

1 JOHDANTO

1.1 Tavoitteet

Tämän opinnäytetyön tarkoituksena oli päivittää SIP-Viihde Oy:n Kauhajoen toimipisteen vanha videovuokrausjärjestelmä uudemmissa käyttöjärjestelmäympäristöissä toimivaksi. Tavoitteena oli tehdä ohjelmasta mahdollisimman käyttäjäystävällinen sekä tuoda siihen uusia asiakkaan toivomia ominaisuuksia kuin myös poistaa siitä käyttämättömiä, turhaan resursseja vieviä toimintoja.

Syksyllä 2009, kun ohjelmiston toteutus oli vielä pahasti kesken, yritys ilmoitti avaavansa uuden toimipisteen Kurikkaan, johon uusi ohjelmisto olisi välttämätöntä saada toimintaan. Uuden toimipisteen tietokoneisiin oli järjestöntä asentaa useita vuosia vanhaa Windows-käyttöjärjestelmän versiota ainoastaan ottaakseen käyttöön vanha videovuokrausohjelmisto, joka ei kuitenkaan olisi pitkään käytössä vaan uuden Windows-version asennus tulisi lyhyen ajanjakson jälkeen eteen. Tästä syystä ohjelmiston ensimmäinen versio julkaistiin keskeneräisenä ja varsin vajavaisena toiminnoiltaan.

Ohjelman jo valmistuttua mieleen kiiri ajatuksia tietokannan päivittämisestä MySQL-tietokantahallintajärjestelmään sen nopeuden ja monikäyttöisyyden vuoksi. Lisäksi ajatuksena oli toteuttaa verkkosivuille lisäosa, josta asiakas kykenisi suorittamaan tietokantakyselyjä yrityksellä olevista elokuvista. Tämä osio toteutettiin itsenäisenä osana, irti asiakasyrityksestä, mutta ajatuksena on ottaa se myöhemmin yrityksessä käyttöön versiona 2, jos ja kun siihen liittyvät palvelinjärjestelyt saadaan ratkaistua.

1.2 Tietoja asiakasyrityksestä

Viihdestoppi, viralliselta nimeltään SIP-Viihde Oy, on kauhajokinen yritys, joka toimii kahdessa toimipisteessä. Päätoimipaikka sijaitsee Kauhajoella ja syksyllä 2009 avattu uusi toimipiste sijaitsee Kurikassa. Viihdestoppi on perustettu vuonna 1999 ja se työllistää tällä hetkellä noin 5 henkilöä, mukaan lukien omistaja kaksi-

kon. Yrityksen toimialaluokitus on musiikki- ja videotallenteiden vähittäiskauppa, mutta valtaosa yrityksen toiminnasta koostuu videotallenteiden vuokrauksesta.

Viihdestoppi kuuluu Elisa Shopit Piste-ketjuun ja suuri osuus sen liikevaihdosta tulee juuri eri operaattoreiden asiakkaille tehdyistä liittymäsopimuksista. Yrityksellä on edustus lähes kaikista Suomessa toimivista matkapuhelinoperaattoreista. Lisäksi yritys tarjoaa matkapuhelin- ja tietokonetarvikkeita sekä tietokoneiden myyntiä, huoltoa ja mittatilauskasausta.

Kuvassa 1 näkyy Viihdestopin tiloja.



Kuva 1 Viihdestoppi

2 PROJEKTIN KUVAUS

2.1 Yleiskuvaus

Opinnäytetyönä toteutettu ohjelmisto tulee yritykselle vanhan ohjelmiston korvajaksi, joten projekti aloitettiin vanhan ohjelmiston toiminnallisuuden kartoittamisella ja siihen liittyvien epäkohtien kirjaamisella. Päätoimintoja videovuokrausohjelmassa ei ole monta, toiminnan kannalta kaikkein tärkeimpiä ovat ainoastaan vuokraus ja palautus. Perustoimivan ohjelmiston ohjelmointi ei siis suurempia ponnisteluja vaadi. Kyseessä kuitenkin on ohjelmisto, jonka varassa koko vuokraustoiminta on, joten erilaisia toimintoja tukevia ominaisuuksia löytyy paljon. Myös ajan kanssa hioutunut käyttökokemus alkuperäisestä ohjelmasta on huomioon otettavaa, onhan alkuperäinen ohjelma ollut yrityksellä käytössä liki kymmenen vuotta.

Ajatus ohjelmiston tekemisestä lähti yrityksen omistajan toivomuksesta saada ohjelma, joka toimii uusimmilla käyttöjärjestelmillä. Tämä mahdollistaisi sen, että sitä pystyttäisiin ajamaan samalla tietokoneella kuin käytössä olevaa kassajärjestelmää ja tällä tavoin pystyttäisiin minimoimaan yrityksessä käytettävien tietokoneiden määrä. Lisäksi vanhan koneen käyttöjärjestelmän tietoturvan riittämättömyys huolestutti asiakasta.

2.2 Käytettävyyteen liittyvät määritelmät

Opinnäytetyön lähtökohtana oli toteuttaa toiminnoiltaan nopea ja luotettava ohjelmisto, joka korvaisi alkuperäisen ohjelman. Lisäksi asiakkaan taholta toiveena oli minimoida ohjelmassa hiiren käyttö, lähinnä toimipisteen tilanpuutteen vuoksi. Suunnittelussa kiinnitettiin erityistä huomiota toimintojen sulavuuteen ja kursorin oikeaan sijoittumiseen. Kursorin oikea sijoittaminen nopeuttaa ohjelman käyttöä, mutta väärin toteutettuna se hidastaa sekä aiheuttaa ongelmia ohjelman käytössä. Tästä syystä kaikki siirtymiset kenttien välillä on suunniteltu ensin omalla maalaisjärjellä ja vielä sen jälkeen yrityksen työntekijöiden avustuksella.

Ohjelman tulee parhaimmillaan toimia 12 tuntia päivässä ja 360 päivää vuodessa virheettä, joten muistivuotoihin ei ole varaa ja ohjelman tulee myös olla mahdollisimman kevyt. Samaan aikaan kaikki tiedot vuokrauksista löytyvät ainoastaan käytettävissä olevasta tietokannassa, joten sen toiminta tulee olla täysin virheetöntä. Tietokannan toimintaan tulee kiinnittää erityistä huomiota.

2.3 Lisäominaisuudet

Tämän työn pohjana olevaan ohjelmistoon toteutetut lisäominaisuudet liittyvät lähinnä ohjelman graafiseen ulkoasuun, mutta myös toiminnallisia muutoksia tehtiin. Joitain lisäominaisuuksia kehiteltiin muun ohjelmoinnin lomassa, osa toteutettiin asiakkaan toivomuksesta. Kaikki ominaisuudet on ensin hyväksytetty asiakkaalla ennen niiden liittämistä itse ohjelmaan. Opinnäytetyönä kehitettyyn ohjelmaan tuli kuitenkin muutamia ominaisuuksia, jotka eivät aivan sellaisenaan ole yrityksen käytössä vaan vaativat erillisten ohjelmien asentamista.

3 KÄYTETYT OHJELMISTOT JA TEKNIIKAT

3.1 Ohjelmointiympäristö

Ohjelmointiympäristönä toimii Eclipse Galileo SDK 3.5.1, joka on avoimen lähdekoodin lisenssillä kehitettävä ympäristö ja se tukee tässä työssä käytettävää Java-ohjelmointikieltä. Muita tuettuja ohjelmointikieliä ovat esimerkiksi C, C++ ja PHP.

Eclipse valittiin työn kehitysympäristöksi sekä sen laajan Java-ohjelmointikielen tuen että hyvien graafisen suunnitteluominaisuuksien vuoksi. Valintaan vaikutti myös ohjelmoijan aikaisempi kokemus ohjelmasta sekä sen ilmainen ladattavuus Eclipsen kotisivuilta. Lisäksi sen saa hyvin pienellä vaivalla yhteensopivaksi MySQL Server 5.1-tietokantapalvelimen kanssa.

Ohjelmiston lisäominaisuudeksi kaavailtiin verkkosivuille lisättävä tietokannasta hakuominaisuus. Ominaisuuden ohjelmointikieleksi valittiin PHP sen nopean sivun päivityksen vuoksi. PHP-koodin kirjoittaminen ei itsessään vaadi mitään erityistä ympäristöä vaan sitä pystyy tuottamaan millä tahansa tekstieditorilla. Tässä työssä käytössä oli NotePad++, koska siinä on valmis tuki useille eri ohjelmointikielille, myös PHP-kielelle. Tuki ohjelmointikielelle tarkoittaa sitä, että editori pystyy tunnistamaan eri ohjelmointikielet ja ilmaisee sen eri osiot eri väreillä. Se helpottaa esimerkiksi erottamaan PHP-koodin HTML-koodin keskeltä.

3.2 Ohjelmointitekniikat

Opinnäytetyö toteutettiin Java-ohjelmointikielellä. Se on Sun Microsystemsin kehittämä olio-ohjelmointikieli, joka on C++:n pohjalta rakennettu lausekieli. Kehitysvaiheessa C++-kielestä poistettiin kaikenlaista turhaa tarkoituksena saada aikaiseksi kevyt ja turvallinen, käyttöjärjestelmästä riippumaton ohjelmointikieli ja tämän kehityksen lopputuloksena syntyi Java. Java-ohjelmointikielen tyypillisiä piirteitä ovat muun muassa turvallisuus, automaattinen roskien keruu ja web-ohjelmointituki. Esimerkiksi väärän muistialueen käyttäminen ja sitä kautta on-

ohjelmien aiheuttaminen on Java-kielessä mahdotonta mikä tekee siitä turvallisen ohjelmointikielen. Myös tietoturva on ajateltu Javaa kehitettäessä, sillä ohjelmaa ajettaessa virtuaalikone tekee suoritettavalle koodille paljon tarkistuksia. Tämän lisäksi virtuaalikone voi asettaa suoritettavalle ohjelmalle ns. hiekkalaatikon, jossa ohjelma toimii ilman luvatonta pääsyä käyttäjän tiedostoihin. /5/

Javan kehittäminen tapahtuu JDK-kehitysympäristössä, joka on vapaasti ladattavissa Sun Microsoftin web-sivustolta. Samaiselta sivustolta on myös ladattavissa JRE, jossa on vain ja ainoastaan Java-ohjelmien ajamiseen tarvittavat virtuaalikone ja web-ohjelmien ajamiseen vaadittava selaimen asennettava Java Plug-In. Opinnäytetyönä tehdyn ohjelman kehityksessä käytössä oli sekä JDK:n että JRE:n versio 6 jossa päivitys 17 (versionimeltään 1.6.0_17).

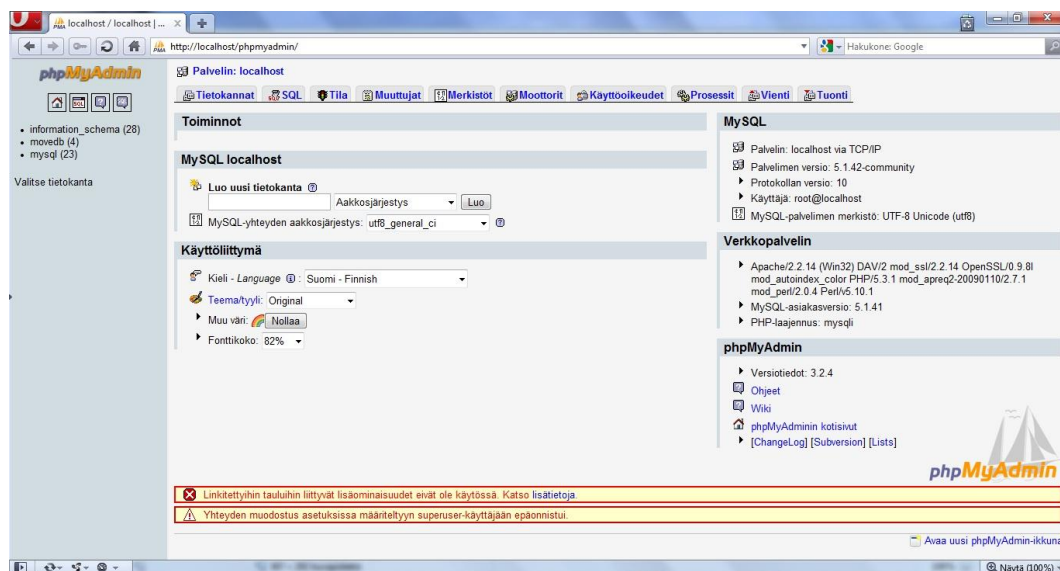
Monesta muusta ohjelmointikielestä poiketen, Java-kielistä ohjelmaa ei käännetä suoraan konekieliseksi, vaan lähdekoodi käännetään ensin tavukoodiksi, jota ajetaan tulkilla, eli JVM-virtuaalikoneella. Juuri tähän virtuaalikoneeseen on ohjelmoitu ne eri kokoonpanojen ja käyttöjärjestelmien piirteet, jotka aikaansaavat ohjelmien käyttöjärjestelmäriippumattomuuden. /5/

Verkkopohjainen lisäominaisuus toteutettiin PHP-kielellä, jota yleisesti käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen luomiseen. PHP-ohjelmointikielen komentoja voidaan kirjoittaa suoraan HTML-sivujen sisään ja tästä syystä sitä voidaan käyttää useilla eri alustoilla ja käyttöjärjestelmillä. PHP on komentosarjakieli, jossa ohjelmakoodi tulkitaan vasta ohjelman suoritusvaiheessa. Tulkittava kieli tarkoittaa sitä, että ohjelmakoodin kääntämisen sijaan ohjelmakoodi suoritetaan rivi kerrallaan. Näin tehtynä ohjelman suorittaminen on hitaampaa kuin samasta koodista käännetyn konekielisen ohjelman suorittaminen, koska tulkin tulee ensin analysoida lähdekoodi ja sen jälkeen vielä suorittaa se. Toisaalta kuitenkin sekä ohjelmakoodin kääntäminen että suorittaminen on lähes yhtä hidasta kuin koodin tulkkaus rivi kerrallaan. /2/

Vaikka palvelin joutuu suorittamaan ohjelman koodin usein, on verkkosivujen toiminta tästä huolimatta suhteellisen nopeaa. Tämä on mahdollista, koska PHP voi toimia Apache-palvelimen sisäisenä moduulina, jolloin se on kiinteä osa WWW-palvelinta ja erillisen tulkin käynnistämistä ei tarvita. /2/

3.3 Tietokannat

Tietokantojen hallintaan valittiin MySQL Server 5.1.42, relaatiotietokantojen hallintajärjestelmä, sen nopeuden ja verkkokäyttömahdollisuuksien vuoksi. Alun perin MySQL:n hallintaan käytettiin komentoriviä ja tekstipohjaisia asiakasohjelmia. Nykyään on myös käytössä esimerkiksi suosittu phpMyAdmin-ohjelma (Kuva 2), joka on selaimella ajettava tietokannan hallintaohjelma. MySQL kehitettiin käytettäväksi verkkosovelluksissa ja kehityksen tuloksena siitä saatiin nopea, laajennettava ja helposti ylläpidettävä tietokantojen hallintajärjestelmä. Lisäksi sen ominaisuuksiin kuuluu siirrettävyys ja riippumattomuus ohjelmointikielestä, mikä mahdollistaa sen monikäyttöisyyden.



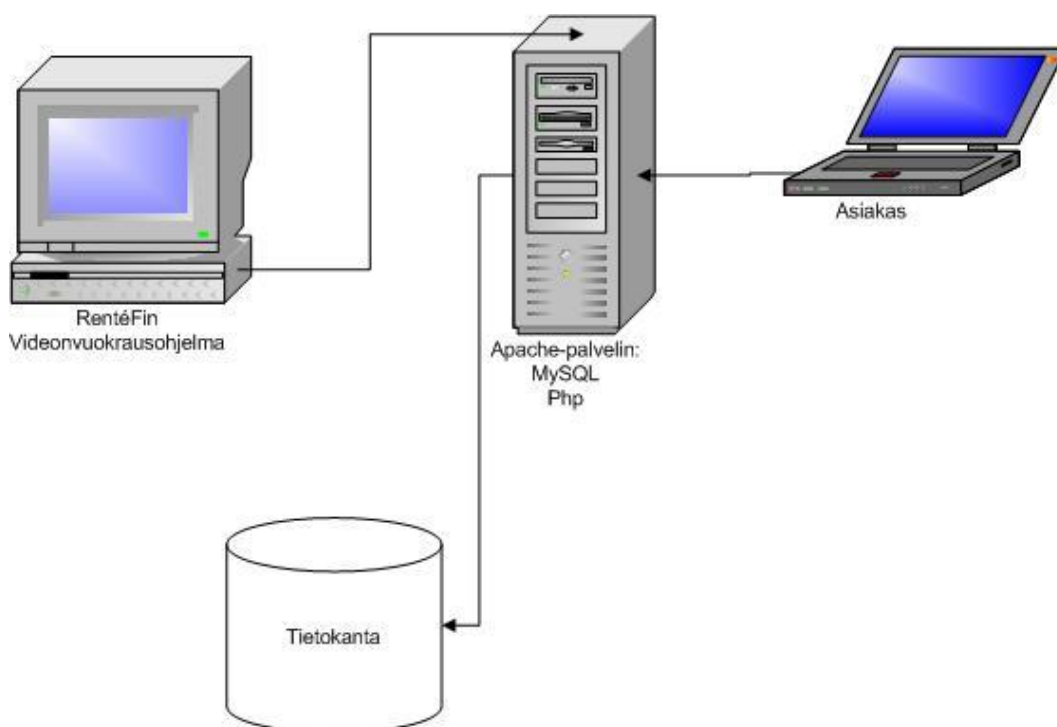
Kuva 2 PhpMyAdmin ohjelmasta.

Java-ohjelmointikielellä yhteys MySQL-relaatiotietokantojen hallintajärjestelmään otetaan, kuten muihinkin vastaaviin hallintajärjestelmiin, salasanan ja tunnuksen avulla. Jokaisella ohjelmointikielellä on lisäksi omat komentonsa ja kirjas-

tonsa tämän yhteyden luomiseen. Yksinkertainen rakenne on yhteyden muodostus, itse kysely ja siihen vastauksen saaminen. Kyselyt annetaan SQL-nimisellä kielellä, joka on teollisuusstandardi. /3/

3.4 Palvelin

Verkkosivuille asennettavaan Php-ohjelmointikielellä toteutettua lisäosaa varten palvelinympäristöksi valittiin Apache HTTPd Server 2.2.14 -palvelin. Se on avoimeen lähdekoodiin perustuva HTTP/web-palvelinohjelma. Apache-palvelin on yksi yleisimmistä palvelinratkaisuista, ehkäpä juuri sen ilmaisuudesta johtuen. Sen suosion taustalla ovat erityisesti nopeus ja toimintavarmuus. Lisäksi se on saatavilla useimmille eri käyttöjärjestelmille ja jos sitä ei vielä ole käännetty haluamaasi ympäristöön, avoimen lähdekoodin ansiosta sen voi itse siirtää haluamalleen alustalle. /1/



Kuva 3 Palvelinympäristö

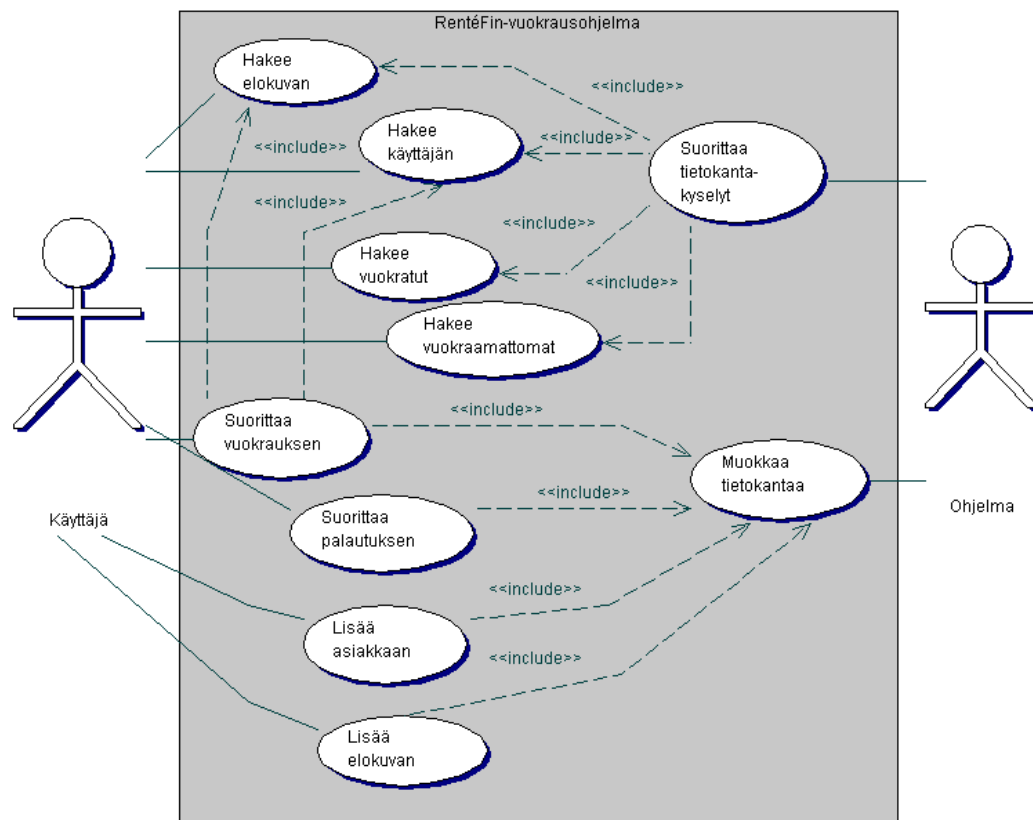
Kuvassa 3 on esitetty opinnäytetyön palvelinympäristö. Samalla palvelinkoneella, Apachen kanssa, sijaitsee MySQL-tietokantojen hallintajärjestelmä, koska

verkkosivuille asennettava lisäosa hakee tietoja tietokannasta. Yksinkertaistettuna palvelin toimii siten, että asiakaskone lähettää pyynnön palvelimelle, joka sitten esimerkiksi lähettää pyyntöä vastaavan verkkosivun asiakaskoneen selaimelle. PHP-koodilla kirjoitettu verkkosivu ei kuitenkaan ole aivan näin yksinkertainen. Se asettaa käytettävälle palvelimelle tiettyjä vaatimuksia. Asiakaskoneen pyytäessä palvelimelta PHP-sivua, ei palvelin voi vain lähettää PHP-koodia sisältävää tiedostoa asiakaskoneelle sellaisenaan, vaan se joutuu ensin suorittamaan tiedoston sisältämän koodin, aivan kuin ajamaan ohjelman ja lähettämään vasta valmiiksi ajetun tiedoston. /2/

4 TOIMINTOJEN KUVAUS

4.1 Yleiskuvaus

Opinnäytetyössä toteutettu ohjelma pohjautuu täysin alkuperäiseen ohjelmistoon ja suurin osa käytössä olevista toiminnoista on sieltä peräisin. Joitain toimintoja on poistettu asiakkaan toivomuksesta ja joitain toimintoja on lisätty. Kuvassa 4 on esitetty ohjelman perustoiminnot käyttötapaaviona.



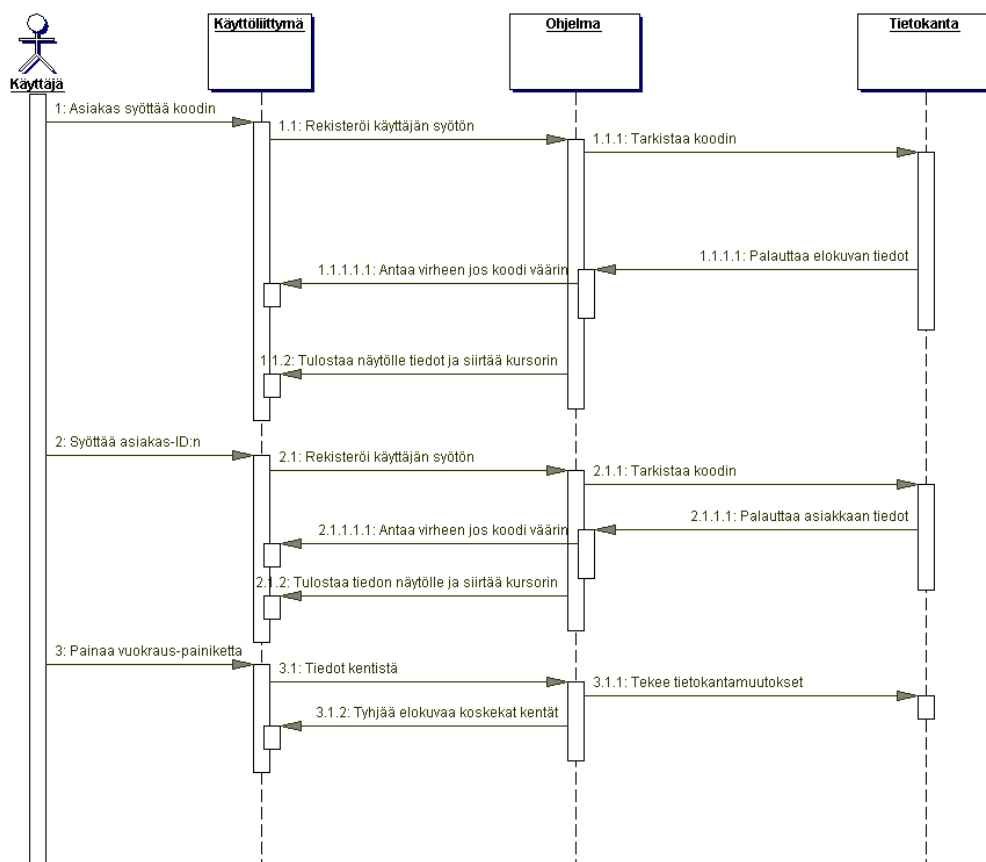
Kuva 4 Käyttötapaavio

4.2 Toiminnot

4.2.1 Vuokraus ja palautus

Ohjelman tärkeimpiä ominaisuuksia ovat vuokraus ja palautus. Tästä syystä näiden ominaisuuksien toiminnallisuuden toteuttamiseen on käytetty määrällisesti eniten aikaa ja lisäksi niiden suunnittelu on tehty tiiviissä yhteistyössä yrityksen henkilöstön kanssa.

Vuokrausta suunniteltaessa huomio keskittyi ensisijaisesti sen toiminnallisuuteen ja käytettävyyteen liittyviin seikkoihin. Kyseessä kun on eniten käytetty ominaisuus koko ohjelmassa, tulee sen toimia moitteettomasti. Tästä syystä ohjelma on kehitetty toimimaan niin sanotusti kaksisuuntaisesti. Vuokraus voidaan aloittaa joko hakemalla ensin vuokrattavan elokuvan tai vaihtoehtoisesti ensin vuokraajan tiedot tietokannasta. Suoritettiin vuokraus sitten tavalla tai toisella, ohjelma itsessään pitää huolen toimintojen sujuvuudesta. Kuvassa 5 on esitetty sekvenssikaa-viona vuokraustapahtuma. Siitä näkee miten ohjelma tekee suurimman työn, ei käyttäjä.



Kuva 5 Vuokraus-tapahtuma

Vaikka ohjelman ikkuna sisältääkin monta kenttää, kuten kuvasta 6 näkee, ei varsinaisesti vuokrauksen kannalta tärkeitä kenttiä ole kuin neljä. Sekä elokuvien, että asiakkaiden hakuun on molempiin varattu kaksi kenttää, joista kuitenkin vain toista tarvitaan vuokrauksen yhteydessä. Tiedot voidaan hakea joko suoraan elokuvan/asiakkaan koodilla tai vaihtoehtoisesti nimellä. Kummallakin tavalla tehtynä, hiiren käyttö on minimoitu ja toiminnon pystyy suorittamaan vain näppäimistön painikkeita hyödyntäen.

Asiakkaiden haku tietokannasta nimellä on toteutettu siten, että asiakkaan pystyy hakemaan syöttämällä hakukenttään ensin osa sukunimestä, jonka jälkeen on mahdollista välilyönnillä erotettuna syöttää osa etunimestä. Näin tehtynä asiakkaiden haku nopeutuu, koska ei ole välttämätöntä syöttää kuin muutamat ensimmäiset kirjaimet asiakkaan etu- ja sukunimestä, ja jo silloin tietokantahaku on löy-

tänyt parhaiten hakuehtoihin sopivat yksilöt. Lisäksi hakutoimintoa on helpotettu niin, että nimen haun jälkeen käyttäjä painaa Enter-painiketta, jolloin kursori siirtyy alla olevaan JList-komponenttiin. Tästä listasta nuolinäppäimiä käyttämällä tai hiiren oikeaa painiketta klikkaamalla, voidaan valita oikean henkilön tiedot ja valinnan jälkeen lähettää kursori elokuvan hakuun. Samankaltainen haku on toteutettu myös elokuvien nimellä hakuun, erotuksena vain on se että elokuvien haussa ei ole mahdollisuutta hakea nimeä kuin yhtenäisenä kirjainyhdistelmänä ja vain elokuvan nimen alusta tai keskeltä.

Vuokrauksen pystyy sulavasti suorittamaan molemmin suuntaisesti. Ohjelma tarkkailee asiakkaan tekemiä toimintoja sisäisillä kuuntelijoilla ja tietyn tapahtuman seurauksena tarkastaa vuokrauksen kannalta oleelliset kentät. Tämän perusteella ohjelma siirtää kursorin sinne missä sen kullakin hetkellä oletetaan seuraavaksi olevan. Eli lyhyesti, jos käyttäjä valitsee ensin asiakkaan tiedot, ohjelma siirtää kursorin elokuvan koodilla haku-kenttään. Kun elokuvan koodi on syötetty, käyttäjän painaessa Enter-painiketta, ohjelma hakee elokuvan tiedot tietokannasta, tämän jälkeen varmistaa, että molemmat tiedot ovat varmasti valittuina ja sen jälkeen siirtää valitsimen vuokraa-painikkeeseen.

Kun vuokraus suoritetaan, ohjelma merkitsee kyseisellä hetkellä vallitsevan päivämäärän ja merkitsee sen tietokantaan vuokrauspäivämääräksi. Palautuspäivämääräksi merkitään normaalisti seuraavan päivän päivämäärä, mutta ohjelmassa on myös mahdollista myöntää vuokra-ajaksi pitempi jakso. Kun vuokraus on onnistuneesti kirjattu tietokantaan, ohjelma tyhjentää kaikki elokuvaa koskevat tekstikentät ja valinnat ja siirtää kursorin valmiiksi elokuvan koodilla haku-kenttään odottamaan seuraavaa vuokraustapahtumaa. Yrityksen edustajan toivomuksesta vuokraajan tiedot jäävät esille, jotta samalle asiakkaalle on mahdollista suorittaa useamman elokuvan vuokraus, ilman tietojen uudelleen hakemista.

Palautus on toteutettu huomattavasti yksinkertaisemmin, koska kyseisen toiminnon tulee tapahtua ilman viiveitä. Palauttamista käsittelevän tietokantakyselyn suorittaminen tapahtuu yhden muuttujan, elokuvan tunnistekoodin, avulla. Jokai-

sella elokuvalla on uniikki tunnistekoodi, joka kenttään syötettynä aikaansaa kyse-
lyn tietokantaan. Kyselyssä etsitään vuokra-aulusta kyseisellä koodilla oleva elo-
kuva, jonka palautettu-kenttä on tyhjä. Jos ja kun vastaavuus löytyy, syötetään
tähän tyhjään kenttään kyseisellä hetkellä voimassa oleva päivämäärä. Samalla
tarkistetaan onko kyseinen elokuva palautettu myöhässä vertaamalla kyseistä päi-
vää siihen tietokannassa olevaan päivämäärään milloin elokuvan palautuksen olisi
pitänyt tapahtua. Jos palautus on tapahtunut annetun määräajan jälkeen, kirjataan
siitä asiakkaan tietoihin pienimuotoinen merkintä huomautuskenttään. Lisäksi yri-
tyksen henkilöstön toivomuksesta palauttamista varten toteutettiin myös erillinen
painike, joka sekä palauttaa kyseisen elokuvan että hakee elokuvan vuokranneen
henkilön tiedot asiakaskenttiin, jolloin kyseiselle henkilölle pystytään ilman vii-
vettä suorittamaan uusi vuokraus.

The screenshot shows a software window titled "Vuokraus". It features a form with the following sections:

- Movie Information:** "Elokuvan koodi:" with "Etsi Koodilla:" and "Uusi kopio:" search fields. A "Tyyllilaji:" dropdown menu and a "Nimi:" text field are also present.
- Customer Information:** "AsiNro:" with "Etsi asiakasnumero:" search field. Fields for "Sukunimi:", "Etunimi:", "Katuosoite:", "Postinumero:", "Postitoimipaikka:", and "Puhelinnumero:" are included.
- Additional Fields:** "Syntymäaika:" with a date picker, "V.I.P.:" checkbox, "Vuokrauksia:" list, "Viiltarikortti:" checkbox, "Vuokralla nyt:" list, and "Sakko:" checkbox.
- Rental Dates:** "Vuokrauspäivä:" with a date range from 27.04.2010 to 28.04.2010 and a numeric input for the number of days (1, 2, 3).
- Buttons:** "Vuokraa", "Vuokraa 3pv", "Palautus", and "Lopetus" are located at the bottom.
- Right Panel:** A vertical stack of buttons: "Elokuvan vuokraukset", "Tallenna muutokset", "Tyhjennä kentät", "Uusi elokuva", "Tallenna muutokset", "Tallenna uusi asiakas", and "Tyhjennä kentät".
- Central List:** A large list area with "Lisää kopio" and "Poista kopio" buttons below it, and "Asiakkaan vuokraukset" and "Poista asiakas" buttons to its right.

Kuva 6 Vuokraus-ikkuna

4.2.2 Tietojen lisäys ja muokkaus

Suurin osa ohjelmassa olevista toiminnoista on sijoitettu kuvan 6 esittämään pääikkunaan, koska kyseinen ikkuna on yrityksessä jatkuvasti auki. Ohjelman yksinkertaistamisen vuoksi asiakkaiden ja elokuvien tietojen lisäys ja muokkaus on toteutettu tapahtuvaksi ikkunassa oleviin tekstikenttiin ja erillisellä painikkeella määrätään tallennetaanko uusi asiakas/elokuva vai muokataanko alkuperäistä. Uuden asiakkaan lisäys on niin ikään toteutettu siten, että hiiren käyttö on minimissään. Kaikissa tekstikentissä on kuuntelija, joka Enter-painikkeen painalluksesta siirtää kursorin seuraavaan tekstikenttään. Viimeisessä tekstikentässä on myös erillinen kuuntelija, joka tarkastaa tehdäänkö muutoksia vanhaan asiakkaaseen vai luodaanko uusi, ja siirtää kursorin oikeaan painikkeeseen. Myös molemmissa painikkeissa on kuuntelija joka tarkistaa, ettei mikään tekstikenttä ole tyhjä, eikä missään ole väärintyyppistä tietoa, kuten tekstiä numeroarvoja vaativissa kentissä. Jos jokin kenttä on tyhjä, ohjelma siirtää kursorin tyhjään kenttään. Vastata kun kaikki tiedot on syötetty, ohjelma tekee muutokset tietokantaan.

Kaikki tietokantamuutokset eivät kuitenkaan ole pelkästään käyttäjän vastuulla. Ohjelma itse tekee myös muutoksia tietoihin, kuten vaihtaa tekstikenttien pienet kirjaimet isoiksi. Myös ohjelmaan asiakkaan toivomuksesta toteutettu sakkolaskuri kirjoittaa automaattisesti asiakkaan tietoihin huomautuksia, jos asiakas on palauttanut elokuvia myöhässä. Tämä ohjelman automaattisesti tekemä huomautus pitää sisällään:

- elokuvan nimen
- päivämäärän jolloin elokuva olisi pitänyt palauttaa
- myöhässäolo päivien lukumäärän
- mahdollisen sakon määrän.

Sakon kaava muodostuu kiinteästä sakosta, joka kerrotaan päivillä jotka elokuva on ollut myöhässä. Kuitenkaan yrityksen edustajan toivomuksesta, sakkoja ei lasketa heti ensimmäisestä päivästä, vaan vasta toisesta päivästä lähtien. Ensimmäisen päivän jälkeen tietoihin tulee vain huomautus siitä, että elokuva on ollut myöhässä, mutta ei sakkoa.

4.2.3 Lisäominaisuuksia

Ohjelman lisäominaisuuksista näkyvimmit ja merkittävimmät olivat niin sanotut käyttäjätasot ja sakko-toiminto. Nämä toiminnot toteutettiin asiakkaan toivomuksesta ja ne olivat suurimmat puutteet alkuperäisessä ohjelmassa. Käyttäjätasojen toteutus on tehty käyttäen apuna jCheckBoxeja, joista tieto siirretään tietokannan yes/no-sarakkeeseen. Asiakkaan tietoja haettaessa käyttäjätason ollessa valittuna, piirtyy käyttäjätasoa varten asetettuun jLabeliin valittua käyttäjätasoa osoittava teksti. Samaista kenttää käytetään myös sakon ilmaisemiseen, tosin silloin kenttään tuleva teksti on punainen (Kuva7). Käyttäjätasot ovat yrityksensä itsensä määriteltävissä ja ne ovat ainoastaan helpottamassa asiakkaiden etujen tunnistusta.

AsNro: 4580 Etsi asiakasnumero:

Sukunimi:

Etunimi: **SAKKO**

Katuosoite:

Postinumero: Postitoimipaikka:

Puhelinnumero:

Syntymäaika:

V.I.P.: Vuokrauksia:

Viihtarikortti: Vuokralla nyt:

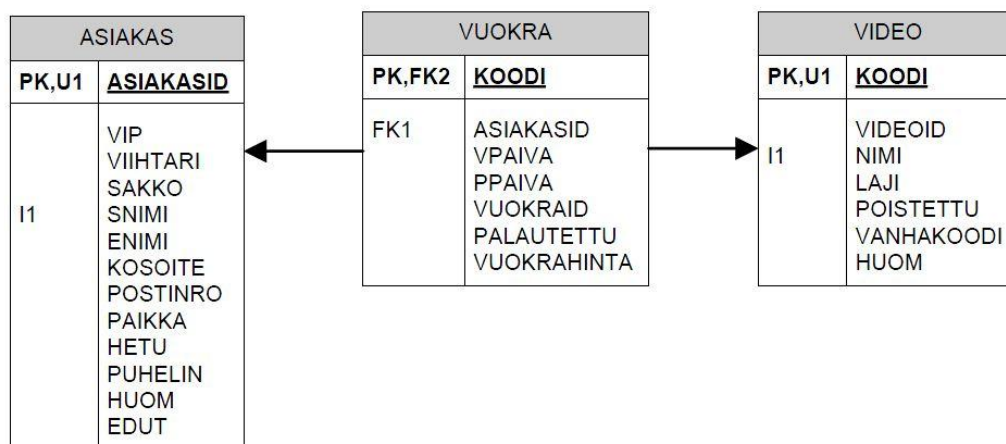
Sakko: Vuokralla nyt:

TESTIVIDEO
VUOKRATTU: 22.04.2010
PALAUTETTU MYÖHÄSSÄ 4
PÄIVÄÄ!
SAKKOA: 6.0EUROA

Kuva 7 Sakkohuomautus

Sakko-toiminto sen sijaan on ohjelman sisäinen ja se aktivoituu kun asiakkaan palauttama elokuva syötetään takaisin tietokantaan ja se on riittävän monta päivää myöhässä. Tämä toiminto estää uudelleen vuokrauksen asiakkaalle, jolla jo ennestään on elokuva vuokralla tai hän on palauttanut sen myöhässä. Vasta kun yrityksen henkilökunta on poistanut aktivoinnin ja veloittanut vuokrauksen myöhässä olosta sakon, asiakas on oikeutettu uuteen vuokraukseen. Tällä estetään myös sakkojen moninkertainen perintä, mikä aikaisemmin oli ongelma. Lisäksi kyseisellä ominaisuudella helpotetaan työntekijän tehtäviä tarkastamalla automaattisesti elokuvan palautus ja kirjaamalla huomautukset elokuvan vuokraajan tietoihin ilman käyttäjän hyväksyntää.

4.3 Tiedot ja tietokannat



Kuva 8 Tietokannan sisältö

Opinnäytetyönä tehdyssä videovuokrausjärjestelmässä käytössä oleva MySQL-relaatiotietokanta sisältää 3 taulua (Kuva 8).

- Asiakas-taulu, joka pitää sisällään kaikki tiedot asiakkaista
- Video-taulu, josta löytyy elokuvaan liittyvät tiedot
- Vuokra-taulu, jossa on vuokralla olevat elokuvat palautuspäivineen.

Tietokannan muuttamisen suhteen mahdollisuudet olivat rajatut, koska alkuperäinen tietokanta oli toiminut yrityksessä jo liki kymmenen vuotta, joten sen sisältämät tiedot olivat korvaamattomia. Muutamien lisäominaisuuksien vuoksi tietokantaan lisättiin uusia sarakkeita. Näitä olivat käyttäjätasot, sakko-toiminto sekä erillinen tekstikenttä mahdollisten etujen ilmaisemiseen. Näitä ovat esimerkiksi liittymän avaamisen yhteydessä mahdollisesti myönnettyt ilmaiset vuokraukset. Osa tietokannassa olevista sarakkeista on todettu olevan ohjelman käytön kannalta turhia ja ne tullaan poistamaan kun niiden sisältämä tieto on tarkistettu ja todettu tarpeettomaksi. Sarakkeiden sisältämä olennainen tieto tullaan siirtämään muihin, käytössä vielä oleviin sarakkeisiin.

Tietokantaan olennaisesti liittyy myös varmuuskopiointi ja se toteutettiin käyttäen hyväksi MySQL-tietokannan hallintajärjestelmän omia työkaluja. Varmuuskopion luonti tapahtuu mysqldump-aliohjelman avulla ja se löytyy MySQLn alikansioista bin. Varmuuskopioinnin käynnistäminen tapahtuu videovuokrausohjelmasta Windowsin komentokehoteen avulla. Java-koodissa on valmiiksi Runtime.getRuntime.exec()-funktio, jonka avulla pystytään käynnistämään erilaisia ulkoisia ohjelmia. Tämän funktion avulla käynnistetään komentokehote ja annetaan sille parametrina mysqldump-aliohjelman sijainti ja sen suorittamiseen vaadittavat arvot. Arvoina aliohjelma vaatii MySQL-hallintajärjestelmän käyttäjätunnuksen, salasanan, varmuuskopioinnin kohteena olevan tietokannan sekä varmuuskopioksi luotavan tiedoston nimen. Alla on esimerkki ohjelman käynnistämisestä toteutettuna Java-ohjelmointikielellä. /4/

```
String command = "mysqldump -uuser -ppassword database -r backup.file";  
Runtime.getRuntime().exec(command);
```

Tiedoston nimellä tai päätteellä ei ole merkitystä, koska palautusohjelma lukee sieltä vain tiedon. Opinnäytetyönä tehdyssä varmuuskopiointiominaisuudessa kuitenkin nimettiin tietokannan varmuuskopiotiedostot selkeyden vuoksi muotoon ddMMyyyy-HHmm.backup. Tiedoston nimessä ensimmäiset luvut kuuluvat päivämäärään ja viimeiset luvut kertovat kellonajan tunneissa ja minuuteissa jolloin

varmuuskopio on luotu. Tiedostojen selkeä nimeäminen helpottaa siinä vaiheessa kun päivässä luodaan useita tietokannan varmuuskopioita. Selkeällä päätteellä erotettuna tiedostot eivät mene muiden tiedostojen kanssa vahingossakaan sekaisin ja tiedoston nimestä näkee suoraan minkä päivän varmuuskopio on kyseessä. Käyttäjä pystyy helposti ylläpitämään tietokannan varmuuskopioita sekä huoletta poistamaan vanhoja tiedostoja suoraan tiedoston nimen perusteella.

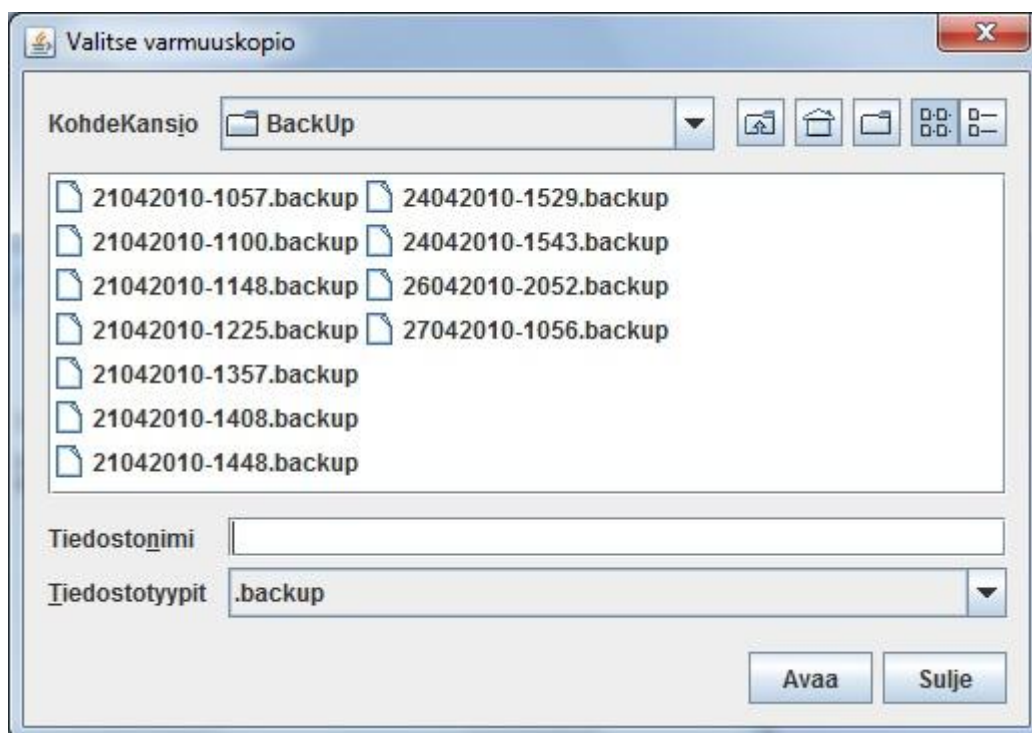
Videovuokrausohjelma käynnistää käyttäjän huomaamatta komentokehotteen ja suorittaa varmuuskopiointi-ohjelman. Ominaisuus toteutettiin käynnistettäväksi sekä manuaalisesti että automaattisesti tiettyyn kellonaikaan tapahtuvaksi. Lisäksi tämä ominaisuuden aikaan sitominen aikaansaa sen, että aina kun ohjelma käynnistetään määritetyn ajankohdan jälkeen, se luo automaattisesti uuden varmuuskopion tietokannasta.

Tietokannan palauttaminen, eli varmuuskopion käyttöönotto, tapahtuu mysql-aliohjelman avulla, joka sijaitsee mysqldump-aliohjelman tapaan MySQL:n alikansiossa bin. Se käynnistetään samalla tavalla käyttäen komentokehotetta. Ainoana erona mysqldump-aliohjelman suorittamiseen on se, että tietokannan palautukselle ei ole erillistä kirjain-parametria, kuten mysqldumpille on ”-r”, joten joudutaan käyttämään suoraan komentokehotteeseen syötettävää nuolikomentoa. Tämä aikaansaa sen, että ajettavan komennon eteen on lisättävä ”cmd /c”, mikä tarkoittaa sitä, että komentokehote ajetaan annetuilla arvoilla ja suljetaan eli aliohjelmaa ei varsinaisesti ajeta Java-ohjelman avulla. Alla on esimerkki tietokannan palauttamisesta Java-ohjelmointikieltä käyttäen. /4/

```
String command = "cmd /c mysql -uuser -ppassword database < backup.file";  
Runtime.getRuntime().exec(command);
```

Arvoina mysql-aliohjelman suorittaminen vaatii MySQL-hallinjärjestelmän käyttäjätunnuksen, salasanan, tietokannan nimen, johon palautus tehdään sekä tiedoston, jossa varmuuskopio sijaitsee. Lisäksi tietokanta, johon palautus tehdään, tulee sisältää valmiiksi taulut ja sarakkeiden sisältämien tietojen tyypit. Ohjelma ei osaa

luoda tyhjästä uutta tietokantaa, koska varmuuskopio-tiedosto sisältää ainoastaan tietokannan sisältämän tiedon, ei sen rakennetta.



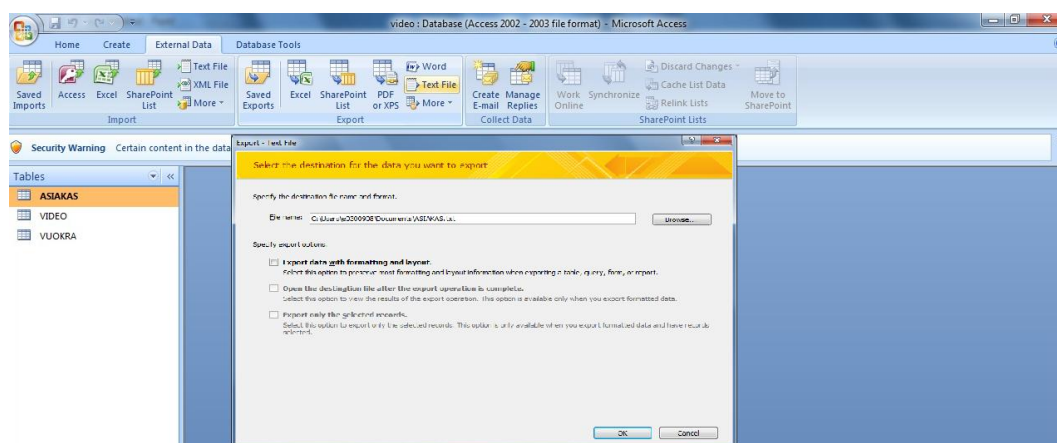
Kuva 9 Varmuuskopion palautus-ikkuna

Varmuuskopion palauttaminen on toteutettu graafisesti käyttäen Javan valmista `JFileChooser`-luokkaa. Se on Windows-tyylinen graafinen kansiodien selausohjelma, joka pienellä muokkauksella saatiin vastaamaan ohjelman tarpeita. Muokkauksina suoritettiin käytettävien ikkunassa näkyvien tekstien ja nappien uudelleen nimeäminen sekä hyväksyttävien tiedosto-päätteiden määrittäminen. Graafisesti toteutettuna käyttäjän on helppo valita listasta haluttu varmuuskopion päiväys. Kuvassa 9 näkyy videovuokrausohjelmassa varmuuskopioiden palauttamisessa käytettävän `JFileChooser`-luokan graafinen ikkuna sekä siihen tehty muokkaus, jossa haettavat tiedostot tulee olla `.backup` muodossa. Valmiissa `JFileChooser`-luokassa pystyy myös määrittämään minkä tiedostokansion se oletuksena avaa. Opinnäytetyönä tehdyssä ohjelmassa varmuuskopiotiedostot sijaitsevat `BackUp`-kansiossa, jonka ohjelma avaa oletuksena.

Käytettävän MySQL-relaatiotietokannan hallintajärjestelmän asentaminen tietokoneella suoritetaan lataamalla MySQL:n verkkosivuilta asennustiedosto sekä erillinen connector-paketti, joka asentamalla tietokanta saadaan yhteensopivaksi Java-ohjelmien kanssa. Itse ohjelman asennus tapahtuu lähes automaattisesti, eikä käyttäjältä vaadita kummempia syötteitä. Oletuksena järjestelmä toimii paikallisen palvelimen portissa 3306.

Yrityksellä alkuperäisenä käytössä ollut tietokanta oli toteutettu Microsoftin Accessilla ja sen sisältämän korvaamattoman tiedon vuoksi opinnäytetyötä varten tietokanta oli välttämätöntä muuntaa MySQL:n käyttämään muotoon. Tähän toimenpiteeseen on myös tarjolla useita ohjelmia, jotka tekevät tietokantamuutoksen automaattisesti. Tässä esitellään kuitenkin toimenpiteen suorittaminen käytössä olleella menetelmällä. Muunto tapahtuu käyttäen Microsoftin Access-ohjelmaa ja MySQL:n omia työkaluja.

Muunnos aloitetaan avaamalla ensiksi alkuperäinen tietokanta Microsoft Accessiin. Tämän jälkeen valitaan koko tietokanta ja Access-ohjelman yläpalkista valitaan ”Ulkoiset tiedot” -välilehti. Tämän välilehden alta valitaan ”vienti” ja sen alta poimitaan tekstitiedosto, kuten kuvassa 10 on esitetty.



Kuva 10 Tietokannan muutos Accessilla

Ohjelma kysyy tallennettavan tekstitiedoston nimen ja kohdekansion jonne tiedosto luodaan. Luotavan tekstitiedoston tulee olla samanniminen kuin tietokannassa

oleva taulu, koska muunto-ohjelma yrittää siirtää tiedoston sisältämät tiedot tauluun, joka on nimeltään sama kuin tiedosto. Seuraavaksi valitaan muutamasta Accessin ehdottamasta vaihtoehdosta haluttu erotusmerkki, joka erottaa sarakkeiden sisältämät tiedot toisistaan. Merkillä ei sinällään ole merkitystä, koska MySQL:n muuntoon tarkoitettussa ohjelmassa pystyy määrittelemään erikseen erotusmerkin. Tässä työssä erotusmerkiksi valittiin tabulaattori, koska erotusmerkinä sitä ei tarvitse erikseen määrittää vaan se on muunto-ohjelmassa oletuksena. Sen huonona puolena voidaan kuitenkin mainita se, että se tekee tekstitiedostosta kooltaan suuren ja tapauksissa, joissa tietokannan resursseja on rajattu, sen käyttö on mahdotonta. Oletuksena Microsoft Access ehdottaa tekstikenttien erottimeksi lainausmerkkejä, mutta tekstitiedostoon kopioituna kenttiä ei tarvitse erikseen erotella. Koska opinnäytetyössä käytetty tietokanta sisältää skandeja, tulee Advancedpainikkeen takaa muuttaa kielen koodaus UTF8-muotoon. Ilman tätä muutosta mysqlimport ei ymmärrä erikoismerkkejä ja jättää ne kirjoittamatta. Riippuen tietokannan koosta ja käytettävän tietokoneen suorituskyvyistä, muunnos Access-tietokannasta tekstitiedostoksi saattaa kestää jopa joitain minuutteja.

Muunnoksen valmistuttua avataan Windowsin komentokehote ja siirrytään MySQL:n alihakemistoon bin. Tekstitiedoston sisältämien tietojen siirtäminen tietokantaan tapahtuu käyttäen MySQL:n omaa aliohjelmaa mysqlimport. Se tarvitsee parametreina muutokseen tehtävät toiminnot, tietokannan nimen johon palautus tehdään sekä tekstitiedoston nimen, jossa palautettavat tiedot ovat. Alla on esimerkki tietokannan muutoksessa käytettävästä lausekkeesta.

```
mysqlimport -uuser -ppassword --delete --local tietokanta asiakas.txt
```

Kyseisessä lausekkeessa parametreina ovat delete, joka tyhjentää taulun ennen kuin sinne lisätään tietoa, sekä local, joka lukee rivit paikallisesti asiakaspalvelimelta. Kun muunnos on mennyt läpi, tiedot löytyvät tietokannasta jonne ne on lisätty. Jokainen tietokannan sisältämä taulu tulee käsitellä samalla tavalla aina tekstitiedostoon muuntamisesta tietokantaan siirtämiseen asti, jonka jälkeen tietokanta on valmis käytettäväksi.

5 TEKNINEN MÄÄRITTELY

Tämän opinnäytetyön tuloksena toteutettu videovuokrausohjelma on ohjelmoitu käyttäen Java-ohjelmointikieltä. Tämän ohjelmointikielen käyttö mahdollistaa sen, että ohjelmiston päivitys ja siirto on helppoa. Käännettynä ohjelma tuottaa ajettavan jar-tiedoston, joka on kooltaan vain muutama sata kilotavua. Sen pystyy kokonsa puolesta vaivatta liittämään esimerkiksi sähköpostin liitteeksi, eikä siirto verkon yli kestä kuin välähdyksen vaikka käytössä olisi hitaampikin yhteys. Ohjelma kokonaisuudessaan vaatii tilaa hyvin vähän ja päivittäminen tapahtuu vain korvaamalla entinen tiedosto toisella, jonka jälkeen ohjelman päivitetty versio on käytettävissä ilman ohjelman erillistä asentamista.

Videovuokrausohjelman käyttö vaatii ainoastaan JRE-ympäristön asennettuna käytössä olevaan koneeseen sekä MySQL-relaatiotietokannan hallintajärjestelmän. Lisäksi erillinen verkkosivulle asetettava tietokannastahaku-ohjelma asettaa erityisiä vaatimuksia käytettävälle palvelimelle. Palvelimesta tulee löytyä tuki Php-ohjelmointikielelle, jotta sivusto toimii oikein.

Suorittavalle tietokoneelle ohjelma ei aseta mitään vaatimuksia. Ohjelma on käyttöjärjestelmäriippumaton ja kevyen rakenteen ansiosta toimii vaivatta hitaammallakin prosessorilla varustetussa koneessa.

6 TOTEUTUS

6.1 Pääohjelma

Opinnäytetyön toteutus alkoi ohjelman ulkoasun hahmottamisella sen jälkeen kun yrityksestä toimitettiin alkuperäisestä ohjelmistosta kuvat. Tarkoituksena oli toteuttaa uusi ohjelmisto mahdollisimman tarkasti alkuperäistä ohjelmaa mukaillen, jotta turhalta käytön opettamiselta vältyttäisiin ja ohjelma saataisiin vähällä vaivalla käyttöön. Graafinen olemus oli siis jo ennalta määrätty, ainoastaan sen toteuttaminen Javalla oli tehtävänä.

Avuksi tuli Eclipse-ohjelma, jossa luokkien luonti visuaalisesti on nopeaa ja yksinkertaista. Valmiita komponentteja löytyy runsaasti ja graafinen hahmottelu onnistuu vaivattomasti hiirtä liikuttaen. Visuaalisen luovuuden rajoittavana tekijänä oli ainoastaan ohjelman käyttöliittymäikkunan koko, joka piti rajata 1024*768 pikseliin. Tämä siitä syystä, että ohjelma toimii samassa tietokoneessa useiden muiden yrityksessä käytettävien ohjelmien kanssa, joita käytettäessä näytön tarkuus tulee olla pieni. Yrityksessä on siis käytössä kosketusnäyttölinen tietokone, joten painikkeiden ja tekstikenttien koko täytyy olla suuri.

Graafista ulkoasua hahmoteltaessa kokeiltiin samalla muutamia alkeellisia vuokrauksen toteuttamismahdollisuuksia, mutta vasta kun alkuperäinen tietokanta saatiin käyttöön, pystyttiin toden teolla paneutumaan ohjelman toimintaperiaatteen. Tästä syntyikin koko ohjelmointiprojektin suurin haaste, keksiä miten edellinen suunnittelija oli tietokantayhteyksien toiminnan ajatellut. Ohjelman toiminta kuitenkin perustuu tietokantaan ja sen toimivuuteen, joten periksi ei saanut antaa.

Tässä vaiheessa aloitettiin ensimmäiset neuvottelut yrityksen kanssa ohjelmiston toiminnallisuuden vaatimusmäärittämisestä. Kaikki liitteen 2 taulukossa esiintyvät ominaisuudet eivät kuitenkaan tulleet selville ensimmäisten neuvottelujen aikana, vaan uusia ominaisuuksia tiedusteltiin ja ehdotettiin aina kun yrityksen toimiloissa vierailtiin. Projektin aikana huomattiinkin miten välttämätöntä ohjelmiston

kehityksen kannalta on pitää tiivistä yhteyttä asiakkaan kanssa jos tarkoituksena on kehittää räätälöity ohjelma. Muutoksia toivomuslistaan saattoi tulla lähes päivittäin tai vaihtoehtoisesti ajatuksia ja ehdotuksia ei saanut yrityksen henkilökunnalta edes pyytämällä.

Koko ohjelmiston kehittäminen oli alkuvaiheessa pelkästään tietokannan tutkimista ja siihen liittyvien toimintojen ja kyselyjen työstämistä. Näistä ensimmäisenä valmistui vuokralla olevien elokuvien haku, jonka toteutus helpotti huomattavasti vuokraus- ja palautus-ominaisuuksien kehitystä. Ei enää ollut välttämätöntä pitää tietokannan hallintaohjelmaa auki ja etsiä sieltä tapahtuneita muutoksia, vaan tietokannassa tapahtuneet muutokset pystyi kätevästi todentamaan suoraan ohjelmasta. Lisäksi tämän tietokannasta haun perusteella pystyi jo hivenen tiedostamaan miten alkuperäinen tietokanta toimi ja mitä tietoja se sisälsi.

```

1  "SELECT VUOKRA.KOODI, VIDEO.NIMI, ASIAKAS.SNIMI, ASIAKAS.ENIMI,
2      ASIAKAS.PUHELIN, VUOKRA.PPAIVA, ASIAKAS.ASNRO, VUOKRA.VPAIVA
3      FROM VUOKRA, VIDEO, ASIAKAS
4      WHERE VIDEO.KOODI=VUOKRA.KOODI
5            AND ASIAKAS.ASNRO=VUOKRA.ASNRO
6            AND VUOKRA.PALAUTETTU IS NULL
7      ORDER BY VUOKRA.PPAIVA, ASIAKAS.SNIMI";

```

Kuva 11 Vuokrattujen elokuvien tietokannasta haku

Alkuperäisen tietokannan rakenteen vuoksi useista tietokannasta suoritettavista kyselyistä tuli melkoisia lausehirviöitä. Tietoja jouduttiin hakemaan useasta eri taulusta ja sarakkeesta. Kuvassa 11 esiintyvä haku on yksi yksinkertaisimmista tietokantaa koskevista kyselyistä. Kyseisessä kyselyssä haetaan tietokannasta kaikki vuokralla olevat elokuvat, niiden vuokraajat tietoineen sekä vuokraus- ja palautuspäivä. Kyselyn lopuksi tietokantakyselyn tulokset järjestetään palautuspäivän ja vuokraajan sukunimen mukaan järjestykseen, eli järjestys päivämäärän mukaan ja samalla päivämäärällä olevat sukunimen mukaan.

Vaikka yrityksen esittämät toiveet ja vaatimukset ohjelmalta olivatkin toisinaan lähes käsittämättömiä, yksikään toivotuista ominaisuuksista ei kuitenkaan jäänyt toteuttamatta. Tietysti osa ominaisuuksista neuvoteltiin pois ohjelmasta jo suun-

nitteluvaiheessa, mutta kaikki luvatut ominaisuudet ohjelmasta löytyy. Myös osa alkuperäisistä ominaisuuksista oli maalaisjärjen vastaisia, eikä työtä tehtäessä ollut varmuutta niiden käyttötarkoituksesta. Yrityksen edustajan kanssa käytyjen neuvottelujen aikana mietittiin eri ominaisuuksien käyttöasteista. Lopulta ohjelman valmistuttua joitain jo valmiiksi tehtyjä ominaisuuksia jouduttiin poistamaan, koska yrityksellä ei kuitenkaan, neuvottelujen lopputuloksesta huolimatta, ollut niille mitään käyttöä.

6.2 Lisäosa verkkosivuille

Verkkosivuille sijoitettavan lisäosan ideoiminen käynnistyi kun MySQL-tietokanta otettiin käyttöön ja se on yksi niistä ohjelman ominaisuuksista joka kehiteltiin itsenäisesti ilman yrityksen edustajan toivomusta. Tietokannan päivityksen johdosta tietokannasta pystyttiin hakemaan tietoa käytössä olevan tietokoneen ulkopuoleltakin. Lisäksi Viihdestopin verkkosivuja varten olevaa palvelinta pystyttäisiin hyödyntämään tehokkaammin sijoittamalla MySQL relaatiotietokantojen hallintajärjestelmä sille. Myös yrityksen edustaja innostui ideasta toteuttaa ominaisuus, joka parantaisi heidän näkyvyyttään ja parhaimmassa tapauksessa toisi lisää asiakkaita runsaan elokuvatarjonnan myötä.

Ajatuksena oli toteuttaa niin sanottu rakennuspalikka, joka voitaisiin helposti asentaa mihin kohtaa yrityksen verkkosivuja tahansa. Pitäisi vain kopioida ominaisuuden ohjelmakoodi ja siirtää se yrityksen verkkosivujen lähdekoodiin siihen kohtaan mihin ominaisuus toivotaan. Ei asennuksia, eikä vaikeaa ylläpitoa. Ainoastaan yksinkertainen ohjelmakoodi, joka lähdekoodiin sijoitettuna mahdollistaisi tietokannan käytön jokaisen asiakkaan koneella. Tästä syystä ohjelmointikieleksi valittiin PHP, koska siinä on hyvät ominaisuudet tietokantayhteyksien luontiin ja tietojen hakuun. Lisäksi se ei vaadi ominaisuutta käyttävien asiakkaiden tietokoneilta erityisiä ohjelmia. Pelkät WWW-sivut avaava selain riittää.

Ominaisuuden kehittäminen aloitettiin miettimällä mahdollisia tietoja joita ohjelman tulisi tietokannasta hakea. Kaikista tietokannasta olevista tiedoista asiakkaan

kannalta olennaisimpia ovat vain elokuvan nimi ja se, onko se vuokralla vai vapaana. Tietokantahakujen toteuttaminen ei kuitenkaan ollut niin yksinkertaista, koska tiedot sijaitsevat useassa eri taulussa ja tästä syystä yhdellä haulilla kaikkien tietojen hakeminen ei ollut mahdollista. Lisäksi kahdella eri haulilla toteutettuna graafinen olemus ei olisi miellyttänyt silmää, koska vuokralla olevat olisivat joko olleet haun tuloksena syntyneen listan ensimmäisenä tai viimeisenä. Lopullinen haku taivutettiin muotoon, jossa haetaan kaikki elokuvat ja kaikki vuokralla olevat elokuvat. Nämä tunnustekoodin mukaan numeeriseen järjestykseen laitettuna tekevät näytölle ilmestyvästä listasta visuaalisesti näyttävän. Liitteessä 1 on kuva lisäosasta kokeellisesti yrityksen sivuille asennettuna.

Haku toteutettiin siten, että tietokannasta noudetaan kaikki elokuvat, joiden nimessä esiintyy haettu sana tai kirjaimet. Toimintoa käytettäessä ei välttämättä tarvitse muistaa haettavan elokuvan koko nimeä vaan riittää kun mielessä on joko kokonainen sana nimestä tai vaihtoehtoisesti kirjainyhdistelmä jostain kohtaa elokuvan nimeä.

Kuvassa 12 on esimerkki toteutetusta ohjelmakoodista. Siinä esitetään tietojen haku tietokannasta ja haettujen tietojen keskenään vertailu. Ensimmäisessä kyselyssä haetaan kaikki elokuvat, joissa esiintyy hakukenttään syötetty teksti ja jälkimmäisessä kyselyssä haetaan kaikki vuokralla olevat elokuvat. Tämän jälkeen hakutuloksina saatuja elokuvan koodeja vertaillaan keskenään ja vastaavuuden löytyessä tulostetaan näytölle teksti vuokralla, päivämäärä jolloin elokuva pitäisi palauttaa ja elokuvan nimi. Jos taas vastaavuutta ei löydy ja kaikki elokuvat on käyty läpi, tulostetaan teksti vapaana, tyhjää päivämäärän kohdalle ja lopuksi elokuvan nimi.

```

$_POST['haku'] = "%".$sana."%";

$query = "SELECT * FROM VIDEO WHERE NIMI LIKE '{$_POST['haku']}' AND KOODI>0 ORDER BY KOODI";
$query2 = "SELECT KOODI, PPAIVA FROM VUOKRA WHERE PALAUTETTU IS NULL AND KOODI>0 ORDER BY KOODI";

$result = mysql_query($query);
$result2 = mysql_query($query2);

while ($row = mysql_fetch_assoc($result)) { $kaikki[] = $row; }
while ($row2 = mysql_fetch_assoc($result2)) { $vuokralla[] = $row2; }

while($laskuri < count($kaikki)) {
    echo "<tr>";
    for($i=0;$i<=count($vuokralla);$i++) {
        if($vuokralla[$i]['KOODI'] === $kaikki[$laskuri]['KOODI']) {
            echo "<td><font color='red'> vuokralla </font></td>";
            $date = strtotime($vuokralla[$i]['PPAIVA']);
            $final_date = date("d.m.Y", $date);
            echo "<td>".$final_date."</td>";
            echo "<td>".$kaikki[$laskuri]['NIMI']."</td>";
            break;
        }
        if($i==count($vuokralla) && $vuokralla[$i]['KOODI'] !== $kaikki[$laskuri]['KOODI']) {
            echo "<td><font color='green'> vapaana </font>";
            echo "<td></td>";
            echo "<td>".$kaikki[$laskuri]['NIMI']."</td>";
            break;
        }
    }
    $laskuri++;
    echo "</tr>";
}

```

Kuva 12 Esimerkki PHP-ohjelmakoodista

Lisäosan lopulta valmistuttua sen konkreettista hyötyä alettiin pohtia ja lopulta tultiin siihen johtopäätökseen, että sen käyttöönotosta saattaisi olla Viihdestopille enemmän haittaa kuin hyötyä. Aikaisemmin, kun yrityksellä ei vielä ollut kilpailijaa samalla paikkakunnalla, verkkosivuilla olevasta hakuominaisuudesta olisi ollut enemmän hyötyä. Nyt tilanne olisi pahimmillaan se, että asiakas tarkastaisi Viihdestopin sivuilta onko toivottu elokuva vapaana ja sen sillä hetkellä ollessa vuokralla, asiakas saattaisi lähteä kilpailevan yrityksen tiloihin etsimään kyseistä elokuvaa. Asiakkaan paikan päällä todetessaan kyseisen elokuvan olevan poissa, hän saattaisi vuokrata jonkun toisen elokuvan tilalle. Lisäominaisuuden käyttämisellä voitaisiin menettää useita yritykselle tärkeitä vuokraustapahtumia.

6.3 Kohdatut ongelmat

Käytössä olleen tietokannan käyttöönotossa kohdattiin koko projektin suurimmat ongelmat. Yksi näistä oli heti kehittämistyön alkuvaiheessa kohdattu tietokantakyselyjen muodostaminen. Toimintojen nopeuden kanssa ei olisi ollut mitään ongelmaa, jos kyselyt olisi voitu suorittaa yhdellä lausekkeella ja teettää suurin työ tietokannalla. Tietokantojen monimutkaisuudesta johtuen, suurin työ jouduttiin teettämään käytettävällä tietokoneella, mikä asettaa käytettävälle tietokoneelle tiettyjä rajoituksia. Aivan vanhimmalla koneella ei ohjelma toimi välttämättä riittävän nopeasti.

Elokuvien haussa ongelmallisinta oli yrityksen toivomus saada elokuvalistaan nimen eteen merkki siitä, onko elokuva vapaa vai vuokralla. Tämän toiminnon toteuttaminen nopeasti oli vaikeaa ja siitä jouduttiinkin tekemään kompromissi. Ohjelma hakee ensin tietokannasta sekä kaikki elokuvat että kaikki vuokralla olevat. Ohjelma vertaa jokaista elokuvan koodia vuokralla olevien elokuvien koodeihin. Vastaavuuden löydyttyä vertailu loppuu ja elokuvan nimen eteen laitetaan miinusmerkki. Jos taas vastaavuutta ei löydy ja kaikki elokuvat on käyty läpi, elokuvan nimen eteen tulee plus-merkki. Toiminto ei ole varsin nopea, mutta tietokoneiden suorituskyvyn kehittyessä se ei ainakaan hidastu. Toisaalta taas tietokannassa olevien taulujen koon kasvaessa läpikäytäviä tietoja on enemmän, joten kehittämissä työ ohjelman puitteissa tulee jatkumaan.

Tietokannan varmuuskopioinnin toteuttaminen aiheutti myös hieman harmaita hiuksia. Yksinkertaisen Access-tietokannan varmuuskopiointi olisi ollut huomattavasti helpompi toteuttaa Java-ohjelmointikielellä, mutta MySQL-tietokannan varmuuskopiointiin täytyi soveltaa MySQL:n omia varmuuskopiointiin tarkoitettuja ohjelmaosioita. Eniten ongelmia aiheuttikin Java-ohjelmointikielellä varmuuskopiointi- ja palautusohjelman käynnistäminen. Kommentoikeuteeseen kirjoitettuna aliohjelmat suorittivat tehtävänsä kuten käsketty, mutta Javalla toteutettuna suoritus vastusti.

Ensimmäinen Java-ohjelmointikieleen liittyvä niin sanottu ongelma oli ajettavan ohjelman käynnistäminen. Koska MySQL-kansio sijaitsee Program Files-kansiossa, ei kansiosijainti suoraan `Runtime.getRuntime.exec()`-funktioon syötettynä aikaansaa kuin virheen. Tämä siitä syystä, että ohjelmakoodia ajettaessa ohjelma tulkitsee välilyönnit koodin päätökseksi ja lopettaa koodin lukemisen siihen. Näin lopullinen kansiosijainti jää ohjelmalle epäselväksi, eikä se kyseisestä kansioista löydä suoritettavaksi määritettyä ohjelmaa. Tämän virheen kiertämiseksi tuli komentokehotteessa ”`dir /x`”-komennolla katsoa millä nimellä Windows tulkitsee kyseisen kansion. Korjaukseksi paljastui komennossa suoritettavan kansion nimeäminen `PROGRAM~1`-nimiseksi.

Yleisesti katsottuna projekti eteni suoraviivaisesti ja uusia ominaisuuksia kehitettiin sitä mukaan kun edelliset ominaisuudet saatiin toimimaan, kuitenkin aloittaen ohjelmointi tärkeimmästä vähemmän tärkeään. Ongelmia kohdattiin, mutta kaikkiin kysymyksiin ja esteisiin kuitenkin löydettiin ratkaisu. Lisäksi opettajien ja opiskelukavereiden suosiollinen avustus edesauttoi suurimpien ongelmien selättämisessä.

7 TESTAUS

7.1 Johdanto

Projekti toteutettiin niin sanotusti TDD:ia käyttäen. Tämä tarkoittaa sitä, että ennen ohjelmoinnin aloittamista määritellään jokin testi jolla ohjelman toiminta varmistetaan. Ensimmäisenä kirjoitetaan yksinkertaisin ohjelmakoodi, joka läpäisee määritetyn testin. Hyväksytysti läpäistyn testin jälkeen voidaan alkaa parantamaan koodin designia ja toiminnallisuutta. Opinnäytetyönä tehdyssä projektissa testausta suoritettiin samanaikaisesti ohjelmoinnin yhteydessä aina uuden ominaisuuden valmistuttua sekä tarpeen tullen toimintojen oikeellisuuden varmentamiseksi. Testaus tapahtui useissa eri tietokoneissa sekä muutamilla yleisimmillä käyttöjärjestelmillä. Onnistuneita testituloksia saatiin jopa Macintoshin Snow Leopard-käyttöjärjestelmällä ohjelmaa suoritettaessa. Samalla vahvistettiin tieto ohjelman käyttöjärjestelmä- ja -ympäristöriippumattomuudesta.

Lopullisen version testaus tapahtui pääosin samassa ympäristössä ja samalla tietokoneella kuin itse ohjelmointi. Muutamia kertoja ohjelmaa testattiin suorituskyvyiltään heikompi tasoisella tietokoneella todellisen nopeuden selvittämiseksi. Lopullisesti ohjelman oikeellinen toiminta pystyttiin kuitenkin todentamaan vasta yrityksen omilla tietokoneilla.

7.2 Ympäristö

Ohjelman toimintaa on testattu pääsääntöisesti Windows XP-käyttöjärjestelmällä, mutta kehittämistyön aikana ohjelman eri versiot ovat vierailleet muissakin käyttöjärjestelmissä. Ohjelmaa on kokeellisesti suoritettu samalla tietokoneella alkuperäisen ohjelman kanssa. Toimenpiteen tavoitteena oli testata uuden ohjelman tietokannan käsittelyn nopeutta ja verrata sitä alkuperäiseen. Muita käytettyjä testausympäristöjä ovat olleet Microsoftin Windows Vista sekä tämän hetken uusin käyttöjärjestelmä, Windows 7.

Alun perin ohjelma tehtiin käytettäväksi Microsoftin Access ODBC-tietokannan kanssa ja valtaosa ohjelman testauksesta on suoritettu käyttäen kyseistä tietokantaa. Opinnäytetyössä käytössä olevan MySQL-tietokannan käyttöönotto ei vaadi ohjelmakoodin muutoksena kuin toisen käytettävän Java-ajurin ja eri osoitepolun johon yhteys otetaan. Toisin sanoen, Access-tietokannalla testattu ohjelma ja sen tietokantakyselyt, toimivat myös MySQL-tietokannan kanssa. MySQL-tietokannan toimivuutta on testattu ainoastaan ohjelmointiympäristössä.

Tietokantojen ja Php-koodilla toteutetun ohjelmaosion testaukseen valittiin ohjelma nimeltään Xampp for Windows. Kyseinen ohjelma on oikeastaan kokoelma useita eri ohjelmia samaan pakettiin pakattuna ja se on ladattavissa valmiina asennuspakettina Xamppin verkkosivuilta. Xamppin kehittämisen ajatuksena on ollut mahdollistaa Apache-palvelimen käyttö kaikille, myös niille jotka kokevat palvelimen asentamisen haastavaksi. Xamppin vahvuuksia ovatkin helppo asennus, sekä monipuoliset ominaisuudet. Se sisältää muun muassa tässä työssä käytettävät MySQL-relaatiotietokantojen hallintajärjestelmän sekä Apache-palvelimen. Testaukseen tämän ohjelman valintaa puolsi myös se, että kyseisessä paketissa kaikki toiminnot ovat jo valmiiksi konfiguroitu, joten ohjelman asentamisen jälkeen pystyttiin nopeasti suorittamaan tietokantoihin ja verkkopalveluihin liittyvät testaukset.

8 YHTEENVETO

Opinnäytetyönä toteutettu projekti polkaistiin käyntiin verkkaisella vauhdilla, koska ohjelmalle ei varsinaisesti ollut asetettu pakollista valmistumisajankohtaa. Kuitenkin kesällä 2009 yritys ilmoitti uuden toimipisteen avaamisesta, joten ensimmäisen, perustoiminnoiltaan toimivan version valmistumiselle tuli kiire. Myös samantyyppinen rutistus koettiin lopullisen version kanssa, kun yrityksen käytössä olleen alkuperäisen ohjelman sisältämä tietokone sanoi sopimuksensa irti. Tästä syystä lopullinen versio tuli saattaa päivänvaloon pikaisesti, joten ohjelman viimeisen version testaus jäi osin asiakkaan vastuulle.

Tätä raporttia kirjoitettaessa opinnäytetyön tuloksena syntynyt kokonaisvaltainen ohjelmisto ei ole vielä yrityksen käytössä, vaan se käyttää Access-tietokantaa hyödyntävää ohjelmaa. Tulevaisuudessa ohjelmisto tullaan ottamaan käyttöön kun palvelinympäristön yhteensopivuus on varmistettu.

Mielestäni projekti onnistui hyvin ja lopputuloksena syntynyt ohjelma on saanut kiitosta sekä käyttäjiltä että ohjelmoijalta itseltään. Kaikki ominaisuudet, joita ohjelman oikea toiminta vaatii, on toteutettu ja ne toimivat kuten kuuluu. Lisäksi opinnäytetyönä toteutuvassa ohjelmassa on paljon ominaisuuksia, joihin jo ohjelmointivaiheessa on osoitettu kiinnostusta asiakasyrityksen suunnalta.

Työ oli mielestäni mielenkiintoinen ja antoisa. Tiivis yhteistyö asiakasyrityksen kanssa oli minulle uutta ja opin paljon asiakaslähtöisestä suunnittelusta. Lisäksi MySQL relaatiotietokantojen hallintajärjestelmän käyttö ja useat Javalla toteutetut ominaisuudet olivat minulle uutta. Projektin aikana ohjelmoijan taitoni kehittyi ja pääsin sisään projektiluontoiseen työskentelyyn.

LÄHTEET

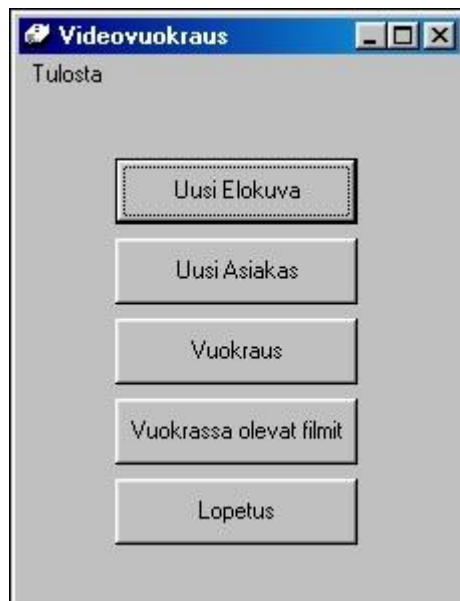
- /1/ Apache HTTP Server Project. About Apache [online] [Viitattu 8.4.2010]
Saatavilla www-muodossa:
<URL:http://httpd.apache.org/ABOUT_APACHE.html>
- /2/ Heinisuo, Rami 2004. PHP ja MySQL: Tietokantapohjaiset verkkopalvelut, 3p. Jyväskylä. Gummerus Kirjapaino Oy.
- /3/ Meloni, Julie C. 2003. MySQL Trainer Kit. Helsinki. Edita Prima Oy.
- /4/ MySQL. MySQL 5.1 Reference Manual [online] [Viitattu 8.4.2010] Saatavilla www-muodossa:
<URL: <http://dev.mysql.com/doc/refman/5.1/en/index.html>>
- /5/ Vesterholm, Mika – Kyppö, Jorma 2004. Java-ohjelmointi, 5p. Jyväskylä. Gummerus Kirjapaino Oy.

LIITTEET

Liite 1 Kuvia alkuperäisestä ja uudesta ohjelmasta

Liite 2 Vaatimusmäärittely

Kuvia alkuperäisestä ja uudesta ohjelmasta



Alkuperäisen ohjelman aloitusikkuna



Uuden ohjelman aloitusikkuna

Videon vuokraus 32-bit

Näytä

Etsi koodi: Uusi kopio: Etsi video:

Koodi: Muut kopiot:

Laji:

Nimi:

Ostohinta: Vuokrahinta: mk

Vuokraukset: kpl

Tallenna videotiedot Lisää kopio Poista kopio

Asiakasnumero: Etsi asiakasnumero: Vuokraukset: kpl, nyt kpl

Etsi:

Sukunimi:

Etunimi:

Katuosoite:

Postinumero:

Postitoimipaikka:

Puhelinnumero:

Henkilötunnus:

Tallenna asiakastiedot Lisää uusi asiakas Tyhjä kentät

Vuokrauspäivä: Vuokra-aika: vrk Palautuspäivä:

Kirjaa vuokraus Palautus Peruuta

Alkuperäisen ohjelman vuokrausikkuna

Vuokraus

Elokuvan koodi: Etsi Koodilla: Uusi kopio:

Tyylilaji: Valitse

Nimi:

Lisää kopio Poista kopio

Elokuvan vuokraukset Tallenna muutokset Tyhjennä kentät Uusi elokuva

AsiNro: Etsi asiakasnumero:

Sukunimi:

Etunimi:

Katuosoite:

Postinumero: Postitoimipaikka:

Puhelinnumero:

Syntymäaika:

V.I.P.: Vuokrauksia:

Viihtikortti: Vuokralla nyt:

Sakko:

Asiakkaan vuokraukset Poista asiakas

Tallenna muutokset Tallenna uusi asiakas Tyhjennä kentät

Vuokrauspäivä:

Vuokraa Vuokraa 3pv Palautus Lopetus

Uuden ohjelman vuokrausikkuna

Koodi	Elokuva:	Asiakas:	Puhelin:	Palautus:
6939	INSIDEMAN *8/06*	VAARANKORPI	*** *****	16.3.2009
8383	PAKO 3.KAUSI *11/08*	TAMSI	*** *****	16.3.2009
8632	MAX PAYNE BLU-RAY *2/09*	YLI-KUUMILA	*** *****	16.3.2009
8162	INSANITARIUM *9/08*	KOIVULA	** *****	16.3.2009
7799	SICK GIRL *2/08*	KOIVULA	** *****	16.3.2009
8590	STEEL TRAP *2/09*	KOIVULA	** *****	16.3.2009
7044	ICE AGE 2 *10/06*	RANTA-OPAS	*** *****	16.3.2009
8329	HANCOCK *11/08*	YLI-VILLAMO	*** *****	16.3.2009
8410	KUNG FU PANDA *12/08*	YLEVÄ	*** *****	16.3.2009
6993	HAMEKYTTÄ 2	LUCÁS	*** *****	16.3.2009
8622	RED SANDS	HIETALAHTI	*** *****	16.3.2009
7578	VIIDAKKOKIRJA *10/07*	PIIKKILÄ	*** *****	16.3.2009
7461	UHKA *8/07*	PIIKKILÄ	*** *****	16.3.2009

Näytä...
 Kaikki Eilen vuokratut Tänään vuokratut Myöhässä olevat

Peruuta Filmejä listassa/vuokralla kaikkiaan: 349/349 kpl

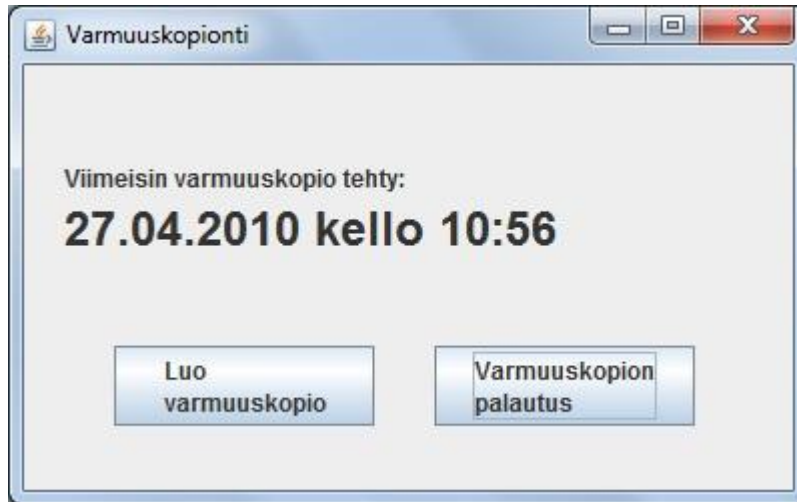
Alkuperäisen ohjelman vuokralla olevien selausikkuna

Koodi	Elokuva	Asiakas	Puhelin	Palautus
8363	TAISTELU TAIVAALLA *11/08*	HYVÄRINEN	*** *****	20.12.2008
8376	SOUTH PARK 8.KAUSI *11/08*	RIIHIMÄKI	*** *****	23.12.2008
7034	NALLE PUH KUMMITTELUA *10/06*	VÄLJÄ	** *****	02.01.2009
6798	NALLE PUH OPPIA IKÄ KAIKKI *3/06*	VÄLJÄ	** *****	02.01.2009
8417	KORKEIN OIKEUS *12/08*	HARJU	*** *****	03.01.2009
8084	CHRYSTALIS *07/08*	LYLY	*** *****	05.01.2009
7124	POLLY WORLD *12/06*	HANNUSAARI	*** *****	12.01.2009
8395	WAR, INC. *12/08*	LÄHTEENMÄKI	*** *****	19.01.2009
6959	VILLI TUSINA 2 *8/06*	HEIKKILÄ	*** *****	25.01.2009
8163	P.S. RAKASTAN SINUA *9/08*	LINDBERG	*** *****	30.01.2009
8473	WANTED *01/09*	PURO	*** *****	15.02.2009
8500	PANKKIKEIKKA *1/09*	WASHINGTON	*** *****	20.02.2009
8443	YÖN RITARI *12/08*	HEIKKILÄ	*** *****	26.02.2009
6864	KAUNOTAR JA KULKURI 2 *5/06*	VÄLJÄ	** *****	02.03.2009
8410	KUNG FU PANDA *12/08*	YLEVÄ	*** *****	16.03.2009
8238	DUUDSONIT 2.KAUSI *9/08*	PAJUMÄKI	*** *****	23.03.2009
8673	SALALIITTO SUDENPESÄSSÄ *3/09*	KALLIO	*** *****	29.03.2009
8179	HOTTIE & NOTTIE *9/08*	PUKKILA	*** *****	06.04.2009
8601	COLLEGE ROADTRIP	KOIVISTO	*** *****	14.04.2009

Näytä **Kaikki** Asiakkaan nimellä:

Poistu Vuokralla: 374kpl

Uuden ohjelman vuokralla olevien selausikkuna



Varmuuskopioinnin aloitusikkuna, näyttää viimeisimmän varmuuskopioajan

VIIHDESTOPPI

- [Verkkokauppa](#)
- [Karaoke](#)
- [Vuokraustoiminta](#)
- [Huolto](#)
- [Uutiset](#)
- [Tietokoneet](#)
- [Elisa Shopit](#)
- [Vihtarikortti](#)
- [Haku](#)
- [Vuokraleffat](#)
- [TV](#)
- [Info](#)

Myymäälä avoinna:
MA-TO 11-20
PE-LA 11-22
SU 14-20

p. 06 2313 044
Puistotie 40
Kauhajoki

info@viihdestoppi.net

Tervetuloa!

[Verkkokauppa](#) > Vuokraustoiminta

Vuokraustoiminta

Vuokraamme liikkeessämme pelejä ja elokuvia.

Voit hakea myymälässämme olevia elokuvia alla olevalla hakukoneella.

tes

	palautus	NIMI
vuokralla	22.04.2010	TESTI *10/2009*
vuokralla	22.04.2010	TESTI 2 *11/2009*
vapaana		TIETOKANTATESTI
vapaana		LAST RITES - TAPPOTUOMIO 599
vapaana		PIRATES OF THE CARIBIAN (DVD) *1/04*
vapaana		TESTAMENTTI (DVD) *7/04*
vapaana		TOPI JA TESSU 2 *8/07*
vapaana		PIRATES OF CARIBEAN 3 *11/07*
		THE HOTTEST STATE *01/08*

Verkkosivujen lisäominaisuus

LIITE 2

Vaatimusmäärittely

1	Elokuvan vuokraus	1
2	Elokuvan palautus	1
3	Uuden asiakkaan lisäys	1
4	Asiakkaan tietojen muokkaus	1
5	Asiakkaan poistaminen	1
6	Uuden elokuvan lisäys	1
7	Elokuvan tietojen muokkaus	1
8	Elokuvan poistaminen	1
9	Vuokraamattomien elokuvien listaus	1
10	Käyttäjätasot	1
11	Sakko-toiminto	1
12	Vuokralla olevien elokuvien selaus	1
13	Yksittäisen asiakkaan vuokraukset kautta aikain	2
14	Yksittäisen elokuvan vuokraukset kautta aikain	2
15	Elokuvan palautus asiakkaan tiedot hakien	2
16	Tietokannan varmuuskopiointi	3
17	Tietokannan varmuuskopion palautus graafisesta käyttöliittymästä	3
18	Tietokannasta raportointi (viikko-, kuukausi- ja vuositasolla)	3
19	Verkkosivujen hakuominaisuus	3